

FIG. 1

Algorithm $E_K(P)$ *// ECB encipher*

```

100 Partition P into  $P_1 \dots P_m$  (where each  $P_i$  is n-bits)
101 for  $i \leftarrow 1$  in  $[1 \dots m]$  do
102    $C_i \leftarrow E_K(P_i)$ 
103 return  $C_1 \dots C_m$ 

```

Algorithm $D_K(C)$ *// ECB decipher*

```

200 Partition C into  $C_1 \dots C_m$  (where each  $C_i$  is n-bits)
201 for  $i \leftarrow 1$  in  $[1 \dots m]$  do
202    $P_i \leftarrow E_K^{-1}(C_i)$ 
203 return  $P_1 \dots P_m$ 

```

FIG. 2

Algorithm $E_K(P)$ // CBC encipher

```

100  Partition P into  $P_1 \dots P_m$  (where each  $P_i$  is n-bits)
101   $C_0 \leftarrow 0^n$ 
102  for  $i \leftarrow 1$  to  $m$  do
103     $C_i \leftarrow E_K(C_{i-1} \oplus P_i)$ 
104  return  $C_1 \dots C_m$ 

```

Algorithm $D_K(C)$ // CBC decipher

```

100  Partition C into  $C_1 \dots C_m$  (where each  $C_i$  is n-bits)
101   $C_0 \leftarrow 0^n$ 
102  for  $i \in [1..m]$  do
103     $P_i \leftarrow E_K^{-1}(C_{i-1}) \oplus C_i$ 
104  return  $P_1 \dots P_m$ 

```

FIG. 3

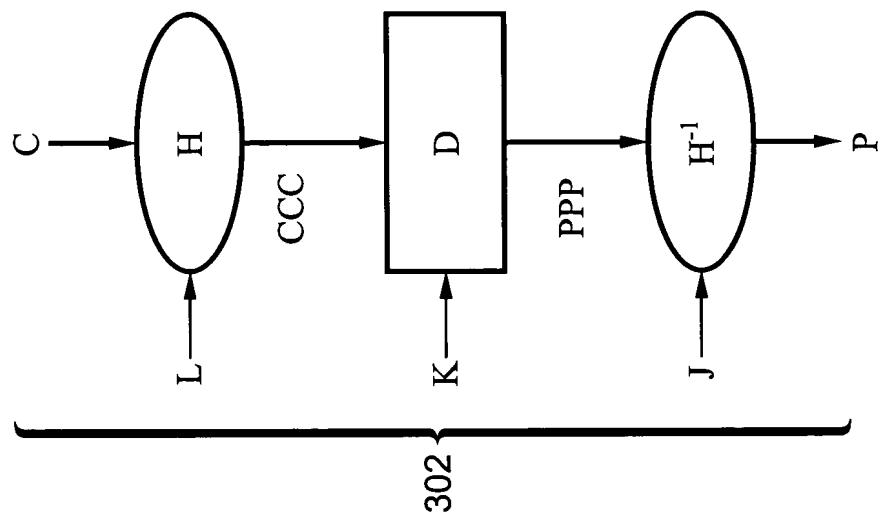
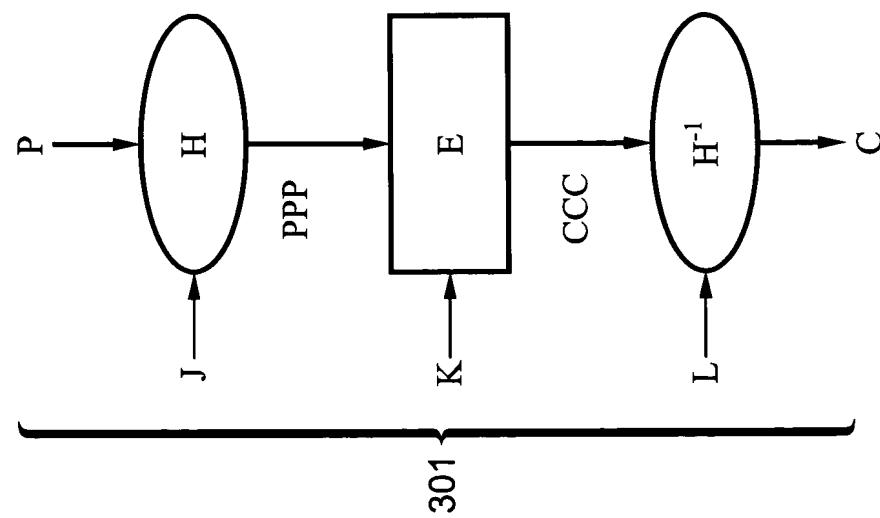


FIG. 4



```
Algorithm double (S) // assumes |S| = 128 and P128(x) = x128+x7+x2+x+1

100  if firstbit (S) = 0 then return S << 1           // left shift
101  else return (S << 1) ⊕ 012010000111           // left shift & xor
```

FIG. 5

Algorithm $E_K(P)$ 100 Partition P into $P_1 \dots P_m$ (where each P_i is n-bits)110 $PPP_0 \leftarrow 0^n$ *// Encipher*111 **for** $i \leftarrow 1$ **to** m **do**112 $PP_i \leftarrow P_i \oplus PPP_{i-1}$ 113 $PPP_i \leftarrow E_K(PP_i)$ 120 $M \leftarrow 2(PPP_1 \oplus PPP_m)$ *// Mask*121 **for** $i \in [1 \dots m]$ **do** $CCC_i \leftarrow PPP_{m+1-i} \oplus M$ 130 $CCC_0 \leftarrow 0^n$ *// Decipher*131 **for** $i \leftarrow 1$ **to** m **do**132 $CC_i \leftarrow E_K(CCC_i)$ 133 $C_i \leftarrow CC_i \oplus CCC_{i-1}$ 140 **return** $C_1 \dots C_m$ **FIG. 6**

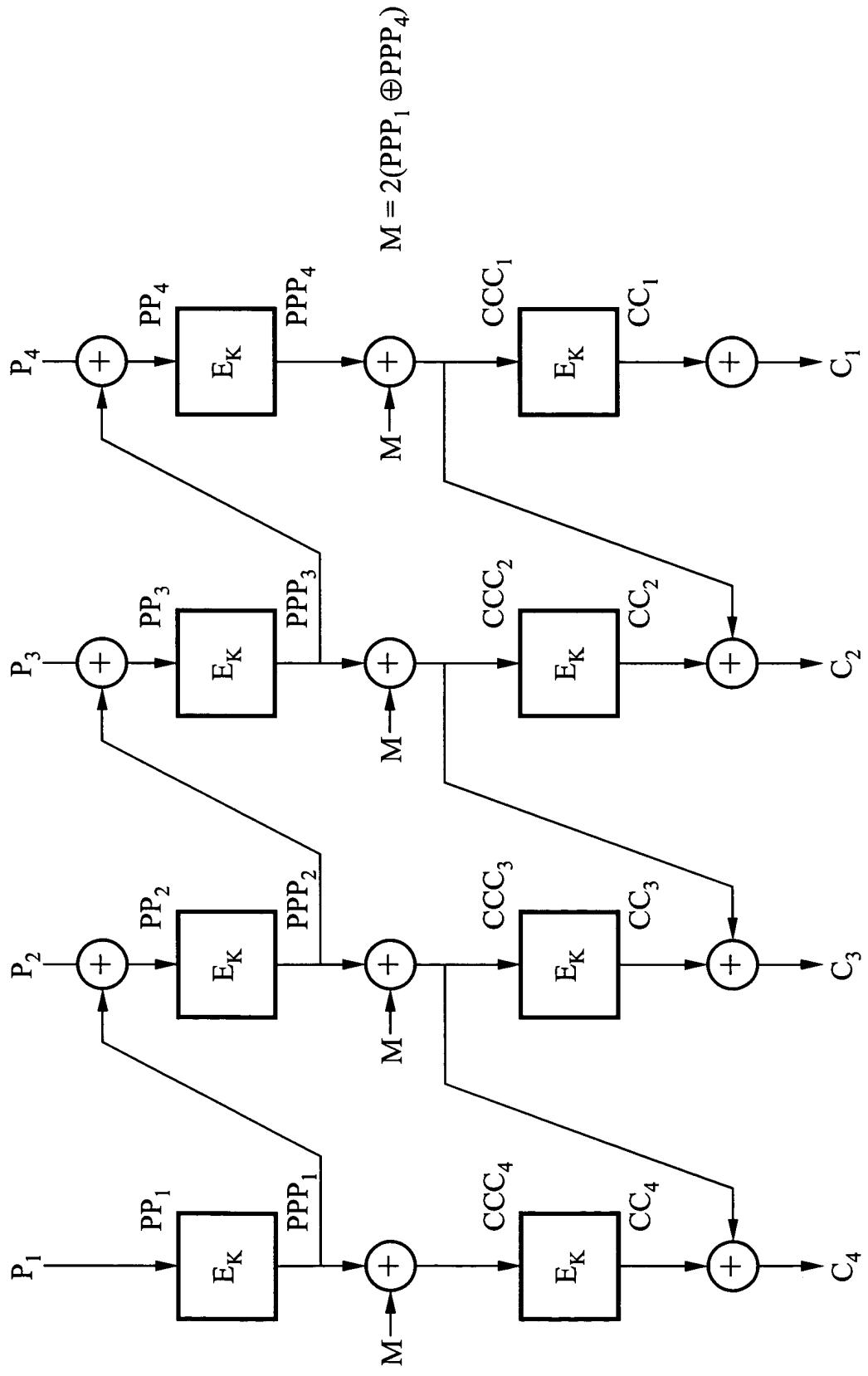


FIG. 7

Algorithm $D_K(C)$

```

100 Partition C into  $C_1 \dots C_m$  (where each  $C_i$  is n-bits)           // CMC decipher

110  $CCC_0 \leftarrow 0^n$                                               // Encipher
111 for  $i \leftarrow 1$  to  $m$  do
112    $CC_i \leftarrow C_i \oplus CCC_{i-1}$ 
113    $CCC_i \leftarrow E_K^{-1}(CC_i)$ 

120  $M \leftarrow 2( CCC_1 \oplus CCC_m )$                                      // Mask
121 for  $i \in [1 \dots m]$  do  $PPP_i \leftarrow CCC_{m+1-i} \oplus M$            // Decipher

130  $PPP_0 \leftarrow 0^n$ 
131 for  $i \leftarrow 1$  to  $m$  do
132    $PP_i \leftarrow E_K^{-1}(PPP_i)$ 
133    $P_i \leftarrow PP_i \oplus PPP_{i-1}$ 

140 return  $P_1 \dots P_m$ 

```

FIG. 8

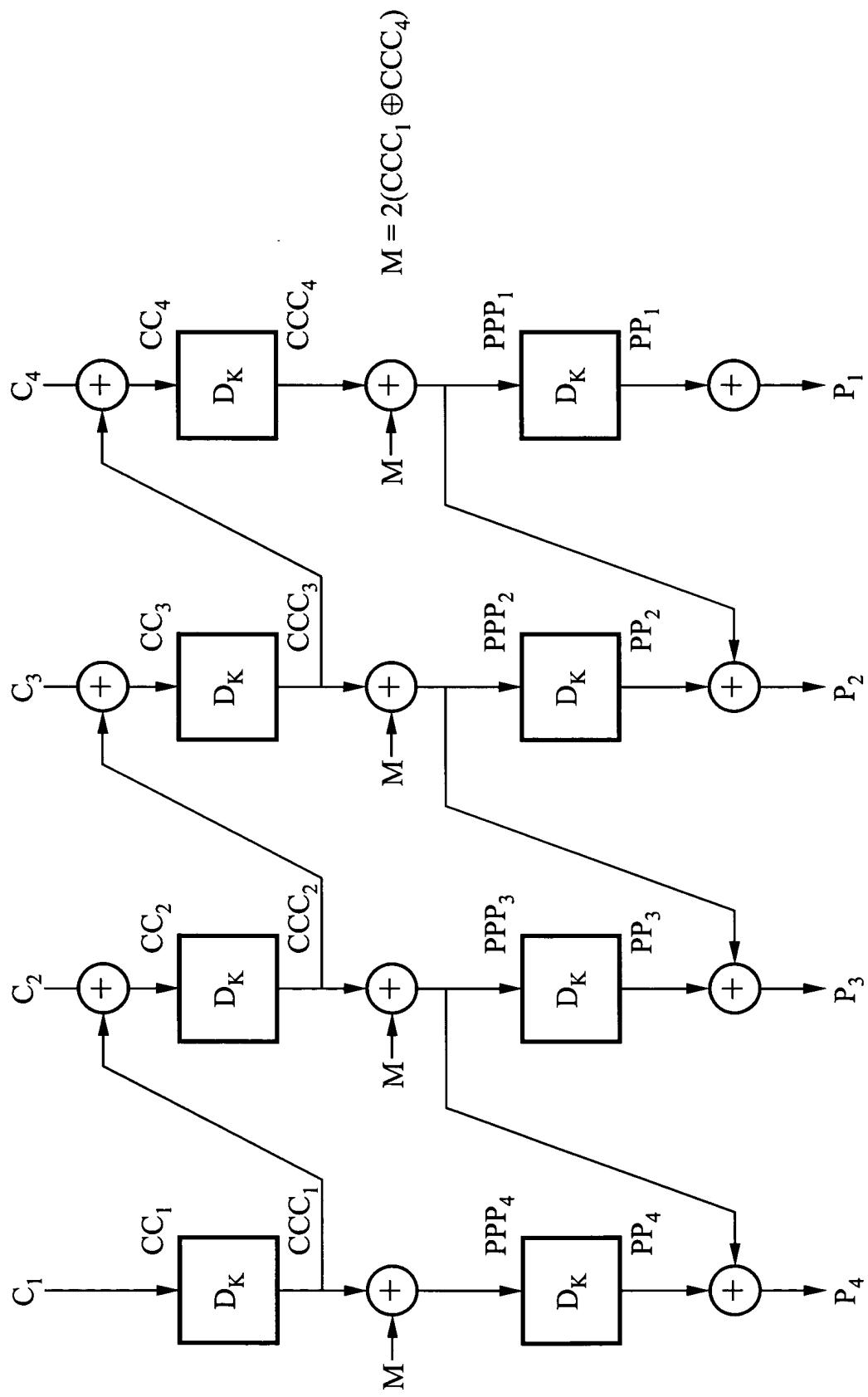


FIG. 9

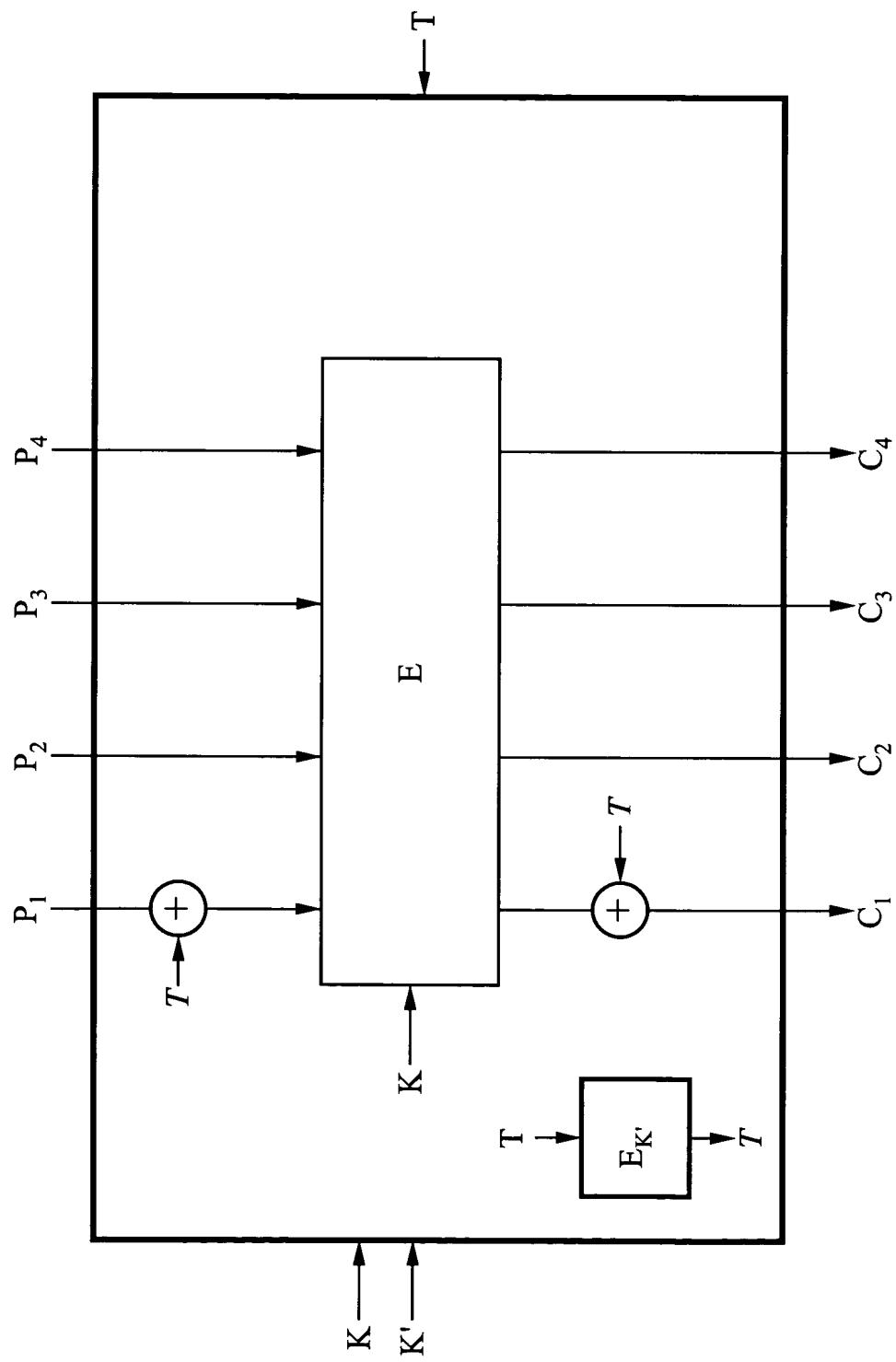


FIG. 10

Algorithm $E_{K'}$ (T, P)100 Partition P into $P_1 \dots P_m$ (where each P_i is n -bits)101 $T \leftarrow E_{K'}(T)$ 110 $PPP_0 \leftarrow T$ *// Encipher*111 **for** $i \leftarrow 1$ **to** m **do**112 $PP_i \leftarrow P_i \oplus PPP_{i-1}$ 113 $PPP_i \leftarrow E_K(PP_i)$ 120 $M \leftarrow 2(PPP_1 \oplus PPP_m)$ *// Mask*121 **for** $i \in [1 \dots m]$ **do** $CCC_i \leftarrow PPP_{m+1-i} \oplus M$ 130 $CCC_0 \leftarrow 0^n$ *// Decipher*131 **for** $i \leftarrow 1$ **to** m **do**132 $CC_i \leftarrow E_K(CCC_i)$ 133 $C_i \leftarrow CC_i \oplus CCC_{i-1}$ 134 $C_1 \leftarrow C_1 \oplus T$ 140 **return** $C_1 \dots C_m$

Algorithm $\mathbf{E}_K^T(P_1 \dots P_m)$ // EME encipher

```

100  L  $\leftarrow$  2EK(0n)                                // Encipher

101  for i  $\leftarrow$  1  $\in$  [1 .. m] do
102    PPi  $\leftarrow$  2i-1 L  $\oplus$  Pi
103    PPPi  $\leftarrow$  EK(PPi)                         // Mask

110  SP  $\leftarrow$  PPP2  $\oplus$  ...  $\oplus$  PPPm
111  MP  $\leftarrow$  PPP1  $\oplus$  SP  $\oplus$  T
112  MC  $\leftarrow$  EK(MP)
113  M  $\leftarrow$  MP  $\oplus$  MC

114  for i  $\in$  [1 .. m] do CCCi  $\leftarrow$  PPPi  $\oplus$  2i-1 M
115  SC  $\leftarrow$  CCC2  $\oplus$  ...  $\oplus$  CCCm
116  CCC1  $\leftarrow$  MC  $\oplus$  SC  $\oplus$  T

120  for i  $\in$  [1 .. m] do                                // Decipher
121    CCi  $\leftarrow$  EK(CCCi)
122    Ci  $\leftarrow$  CCi  $\oplus$  2i-2 L

130  return C1  $\dots$  Cm

```

FIG. 12

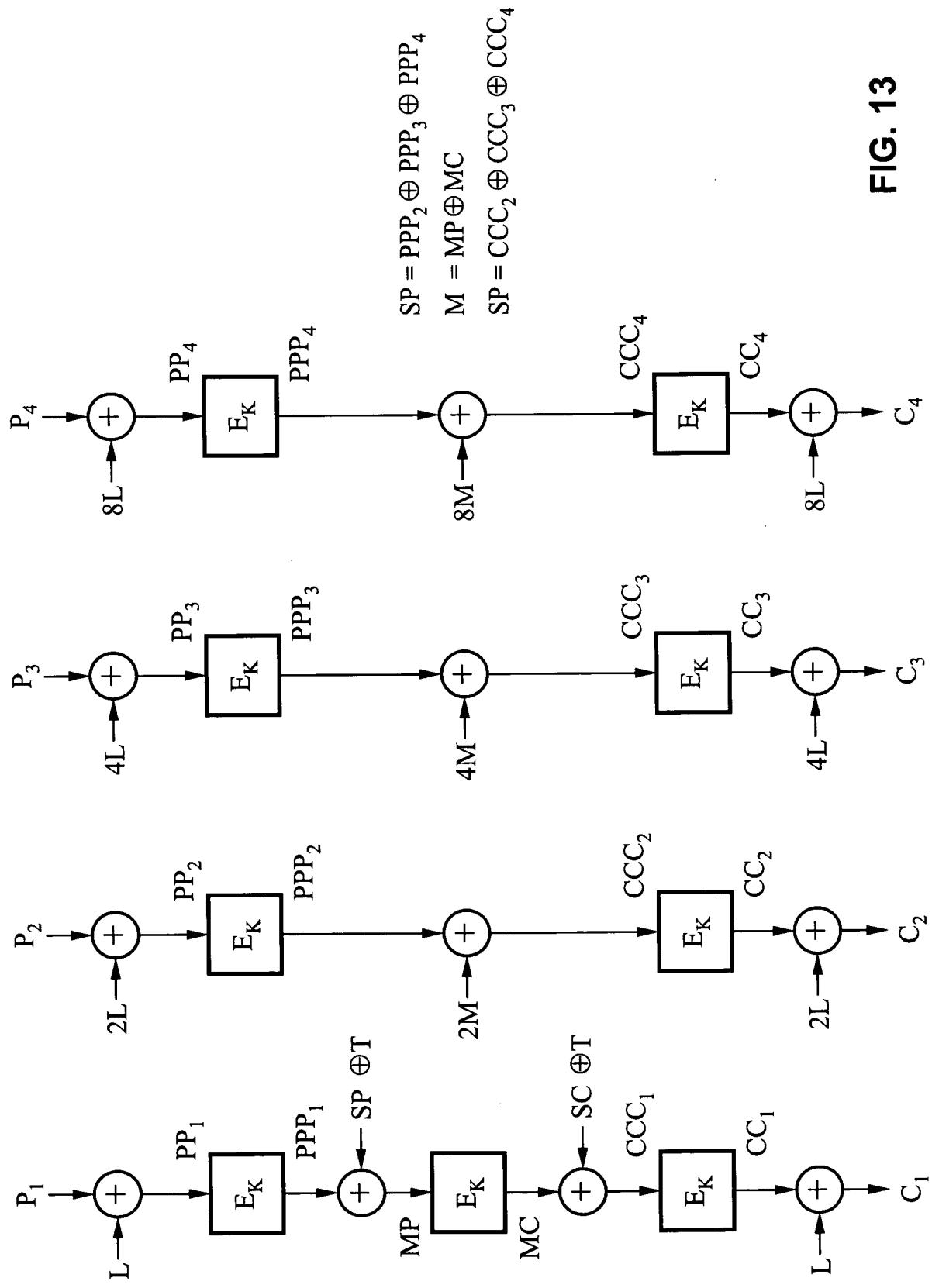


FIG. 13

```

Algorithm  $D_K^T(C_1 \dots C_m)$  // EME decipher
100  $L \leftarrow 2E_K(0^n)$ 
101 for  $i \leftarrow 1 \dots m$  do // Encipher
102    $PP_i \leftarrow 2^{i-1} L \oplus C_i$ 
103    $PPP_i \leftarrow D_K(CC_i)$ 
104    $SC \leftarrow CCC_2 \oplus \dots \oplus CCC_m$  // Mask
105    $MC \leftarrow CCC_1 \oplus SC \oplus T$ 
106    $MP \leftarrow D_K(MC)$ 
107    $M \leftarrow MC \oplus MP$ 
108   for  $i \in [1 \dots m]$  do  $PPP_i \leftarrow CCC_i \oplus 2^{i-1} M$ 
109    $SP \leftarrow PPP_2 \oplus \dots \oplus PPP_m$ 
110    $PPP_1 \leftarrow MP \oplus SP \oplus T$ 
111
112   for  $i \in [1 \dots m]$  do // Decipher
113      $PP_i \leftarrow D_K(PPP_i)$ 
114      $P_i \leftarrow PP_i \oplus 2^{i-2} L$ 
115
116   return  $P_1 \dots P_m$ 

```

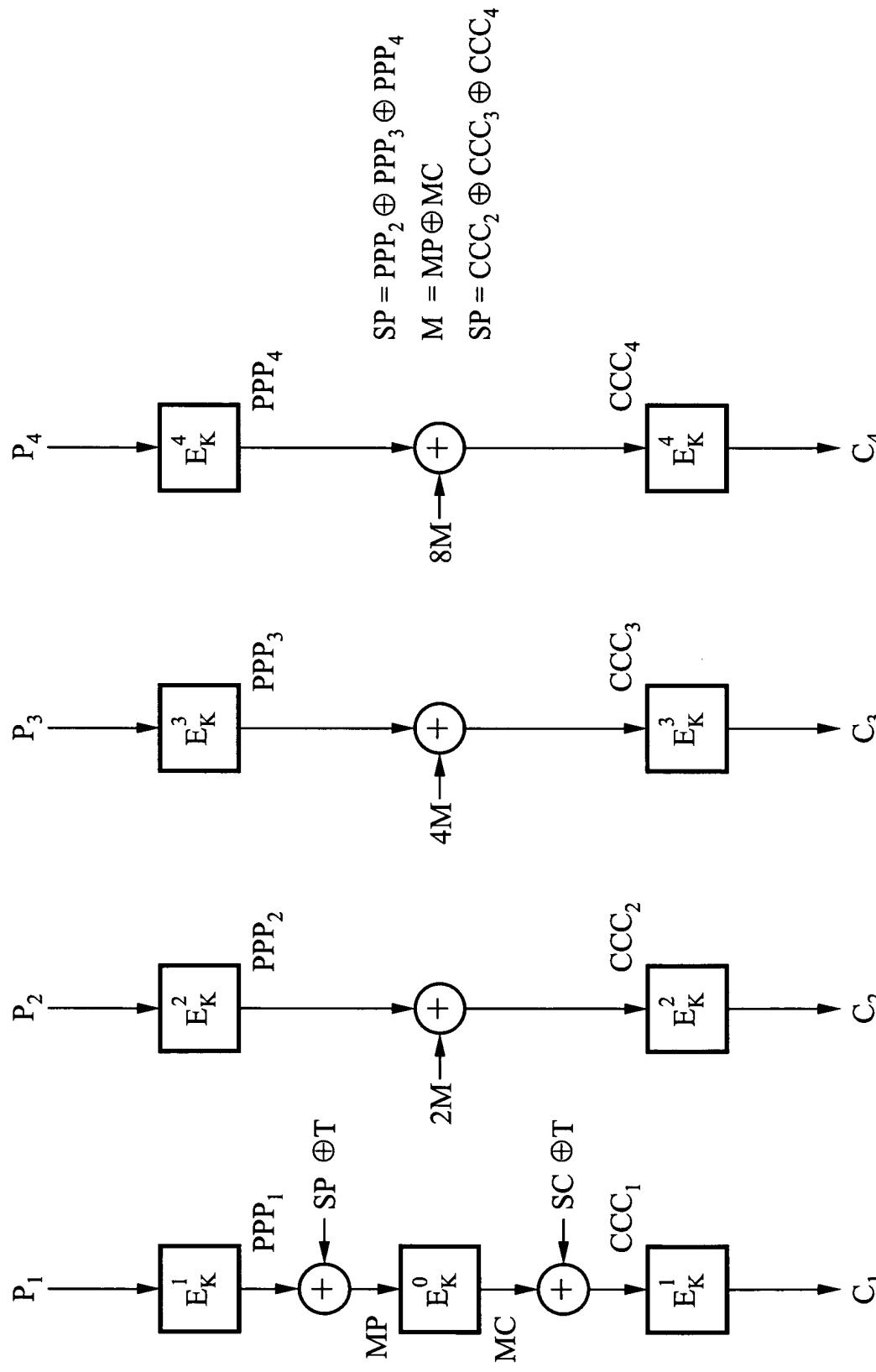


FIG. 15